

Acquisition Knowledge Management with Traceability

Final Report

Funded by the External Acquisition Research Program, Naval Postgraduate School, Contract
N00244-01-C-0034

July 15, 2002

Balasubramaniam Ramesh
Department of Computer Information Systems
Georgia State University
35 Broad Street, Atlanta, GA 30303

Abstract

We view the acquisition of complex systems, products and services a knowledge intensive collaborative activity. We identify problems associated with knowledge management in the context of complex acquisition activities involving cross functional collaborative teams. We map these problems to the characteristics for a knowledge management system to support acquisition activities. We have developed a prototype knowledge management systems to support complex acquisition activities such as the development of acquisition workflows. We have also used the system to manage process knowledge in the management of process fragments in system engineering acquisition activities.. The system can be used to capture and manage tacit and explicit process knowledge involved in complex acquisition activities.

1 Background

Our primary objective in this project is the development of a knowledge management system to facilitate the creation and use of process knowledge documenting **traceability** to critical outputs of an acquisition activity. Knowledge management is an inherently challenging task. The technical challenges include creation of suitable infrastructure to facilitate creation, sharing, transfer, selective dissemination, filtering, and management of explicit process and knowledge [37] [38] as well mechanisms to recognize and access sources of tacit knowledge. In addition, as some tacit knowledge cannot be explicated or codified, providing pathways, channels, and mechanisms for sharing, distributing, and locating tacit knowledge sources is important [39]. We focus on developing such a system centered around the concept of **traceability** defined as **the ability to follow the life of a (physical or conceptual) object, from its origins to its use** [40]. A traceability based process knowledge management system can help create a knowledge network, i.e., a network of people and information systems associated with collaborative, knowledge intensive tasks[41]. Such a process knowledge management system should provide mechanisms for creating, finding, packaging, maintaining, and applying both tacit and explicit knowledge. Specifically, our work addresses the acquisition and use of process knowledge in terms of the following **research questions**:

- **What** process knowledge is represented—representing the semantics of captured knowledge with meta data and salient attributes?
- **Who** are the stakeholders that played different roles in the creation, maintenance, and use of various process knowledge components to provide **pointers to the sources** of tacit knowledge where the limits of codification are reached?
- **Where** is the process knowledge located within the knowledge network—in terms of sources that “document” the knowledge chunks so that it can be accessed using mediators based on the understanding of the capabilities of the source?
- **How** this process knowledge - is represented both by formal and informal means and by how it relates to other knowledge components?
- **Why** a certain process knowledge component was created, modified, or evolved?
- **When** the process knowledge component was captured, modified, or evolved?

As a first step, we have focused on the development of traceability models in the context of supporting an activity critical to the management of defense acquisition, ie., the management of workflow systems used in acquisition activities. Today’s defense organizations must react quickly to changes, rapidly develop new services and products, and at the same time improve productivity and quality and reduce cost. In this process, Business Process Re-engineering (BPR) efforts can help businesses re-design their structures and processes. Among the variety of ways Information technology can be used to support this process, workflow management systems (WFMS) play a central role. The need to coordinate work activities across organizational boundaries (across government organizations and even vendors and suppliers) is increasingly becoming important in the era of e-procurement. Acquisition workflow systems that help manage this process are central to the successful execution of acquisition activities. Our research focuses on supporting the development and maintenance of acquisition activities with traceability knowledge so that acquisition processes can be best understood, acquisition decisions

can be easily explained and acquisition activities can be dynamically reconfigured to meet changing organizational needs and requirements.

2 Traceability in acquisition

Our project is motivated by our recent work on the development of reference models and tools for *requirements traceability* in large-scale systems development [40]. This work demonstrates that the efficiency and effectiveness of traceability as a mechanism for managing complex processes. Extending [42], we suggest that traceability gives essential assistance in understanding the relationships that exist within and across various artifacts produced during the acquisition process. These relationships help establish that trace of the process through which critical acquisition decisions are made and help ascertain how and why outputs of an acquisition process satisfy stakeholder requirements. Following Hamilton and Beeby [43], we view traceability as the ability to discover the history of every feature of the outputs of an acquisition activity so that the impacts of changes in acquisition requirements can be identified. Greenspan and McGowan (1978) state that the ability to allow changes to any artifacts to be traced throughout (the outputs of a complex process such as acquisition) is critical for successful management of the activity. In short, traceability is a characteristic of an acquisition activity in which the requirements are clearly linked to their sources and to the artifacts created by the acquisition process. Specifically, the traceability meta-model derived from [40] and shown in Figure 1 may be used to represent the various dimensions of traceability knowledge we are intended in representing in complex government acquisition processes. Here, our primary focus is the development of models to represent how various **objects; i.e. knowledge fragments** are linked together traceability links. For example, Table 1 shows that the various fragments of knowledge related to a critical acquisition decision that may be represented using a reference model that is derived from the meta-model.

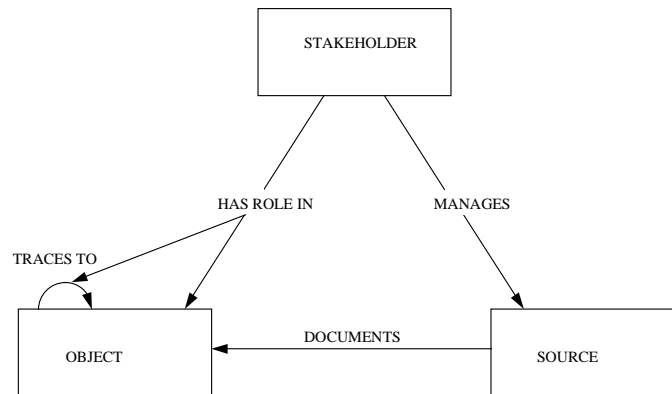


Figure 1: Traceability Meta-Model

<i>Dimension</i>	<i>Example</i>
<i>What?</i>	Rationale for Acquisition Decisions
<i>Who?</i>	System Architect
<i>Where?</i>	In the requirements document Or John Smith, Chief engineer
<i>How?</i>	Using Tool X; Represented as the "Rationale - justifies – Acquisition Decision traceability link
<i>Why?</i>	To facilitate understanding and communication by other acquisition professionals; to avoid rework
<i>When?</i>	At the finalization of the acquisition decision

Table 1: Traceability Dimensions - An example

Formally speaking, a traceability system can be defined as a semantic network in which nodes represent objects (also stakeholders and sources), among which traceability is established through links of different types and strengths. This dependency-directed approach of maintaining consistency of inter-connected objects dates back at least to the work of Stallman and Sussman [44] and is by now well established. The simplest traceability tools are purely relational (i.e. in the form of relational databases or spreadsheets) and do not systematically distinguish different node and link types. Others, such as RDD-100, use a basic entity-relationship structure to distinguish nodes and links, and allow the user to introduce distinctions between different types of nodes and links. However, this begs the question which node and link types *should* be defined to support specialized activities such as acquisition activities.

We propose the use of empirically derived reference models for traceability among objects, stakeholders and sources in acquisition activities. Our extensive studies on theoretical and empirical foundations of traceability practice in system engineering [40, 45], suggests such an approach.

This research will result in the development of reference models comprising the most important kinds of traceability links for various acquisition tasks. Reference models are prototypical models of some application domain, usually organized according to some underlying basic metamodel. The purpose of reference models is to reduce significantly the task of creating application-specific models and systems: the user selects relevant parts of the reference model, adapts them to the problem at hand, and configures an overall solution from these adapted parts. Reference models are therefore an abstraction of best practice, condensed from numerous case studies. They derive their relevance from the slice of practice they cover. The references models developed in this study will be developed within the context of complex acquisition activities to address the research questions identified above. Further, the traceability links so derived can be classified to develop more concrete semantics. With such a well-defined reference model, we can construct and validate a process knowledge management system to support the tasks of various participants in the acquisition process, also specifically addressing the dimensions addressed in the research questions.

3 Workflow Management Systems (WFMS)

Often organizations need to implement large and heterogeneous distributed systems where a set of interrelated tasks can be executed in efficient way. Workflow automation to coordinate activities throughout the enterprise is increasingly recognized as an important approach to support these requirements [1]. The WFMS can also play a vital role in managing knowledge of the enterprise, such as managing business rules [2]. As summarized in [3], the key benefits of WFMS are:

- Improved efficiency - automation of many business processes results in the elimination of many unnecessary steps
- Better process control - improved management of business processes achieved through standardizing working methods and the availability of audit trails
- Improved customer service – consistency in the processes leads to greater predictability in levels of response to customers
- Flexibility – software control over processes enables their re-design in line with changing business needs
- Business process improvement - focus on business processes leads to their streamlining and simplification

Proof of the ever-increasing interest in workflow management is the large number of commercial products that have appeared in the last few years including: Action Workflow System, of Action Technologies; IBM's Flow Mark; Visual WorFlow from FileNet; OPEN/workflow, a WANG's product [4]. Workflow technology has become mainstream application-development tools and application-integration middleware that support the management of business process realization in a variety of application areas. Enterprise Resource Planning Systems (ERP) may be thought of as focusing on business applications of workflow management system [5]. ERP systems may be seen as "a set of more or less fine-grained business functions and a set of predefined workflows that realize complex business operations". When the system is customized for a specific setting, workflows can be modified and new ones can be constructed [6].

In the context of BPR (business process re-engineering), workflow management provides a means for enacting reengineered processes and for gathering information about the actual performance of these processes. As the consequence of rapidly changing environment, businesses need to reengineer business models and processes frequently. Those organizations in the modern business world, which refuse to change, are headed toward rapid obsolescence because they cannot compete. These business realities need to be supported by a Acquisition WFMS by providing the ability to rapidly and dynamically change and enact workflows.

The basic premise of our work is that Acquisition WFMS can benefit from the ability to manage knowledge about how workflow components relate to business process as well as with each other so that the evolution of workflows when the business processes that they are intended to support evolve. Specifically, we advocate the use of traceability, the ability to follow the life of an artifact, as the primary facility to manage the knowledge about the evolution of workflows. In this paper, we present a framework for representing traceability knowledge and discuss tools to support the capture and use of this knowledge to support critical activities in Acquisition WFMS. In the section 2, features of workflow management systems are introduced. We further

discuss the role of knowledge management in supporting workflow management. Section 3 outlines the role of traceability in supporting complex organizational processes. A traceability framework to support workflow design and maintenance is proposed in section 4. Section 5 discusses the features of a knowledge management system to illustrate its functionalities. This is followed by discussion of related work and future work.

Workflow management is defined as the management of processes through the execution of software whose order of execution is controlled by a computerized representation of the process. The primary reason for the popularity of workflow technology is its support for the management trends including reinvention and revitalizing corporations through rightsizing and business-process reengineering. Current implementations of workflow systems such as Enterprise Resource Planning (ERP) systems automate core corporate activities and let companies share common data and practices across the enterprises. Workflow systems are designed to assist groups of people in carrying out work procedures, and contain organizational knowledge of where work flows. Workflow systems are defined as “systems that help organizations to specify, execute, monitor, and coordinate the flow of work items in a distributed environment” [7]. An Acquisition WFMS provides the software tools to define, manage and execute workflows. WFMS have two main functions: a build time function and a run time function. Build time functions enable businesses to model their business procedures and activities, using scripting language. Run time functions help administer workflow process and run time interactions with workflow users and software applications [8].

Most current workflow management systems are static systems, which means to reconfigure WFMSs, systems have to be stopped and re-instantiated. Given the dynamic structure of today's organizations, it is unlikely that business processes are modeled once and executed without any change. More flexibility is desired in WFMSs. The adaptive or dynamic workflow systems are regarded as efficient solutions to the problem [9], [10]. Dynamic Workflow Management Systems (DWFMS) allow users to change the data and the structure of the workflow on the fly. System evolution is unavoidable because business processes evolve continuously caused by internal organizational reforms and external environment changes. The static Acquisition WFMS cannot be efficient in such environments. There is an increasing demand for dynamic adaptive workflow management systems, which can deal with dynamic changing.

Management of the dynamics of workflow configurations is required in a variety of application domains. Consider a manufacturing workflow system that can process orders, schedule processes, control material orders, calculate bills of material, and track orders and products. Any change in manufacturing process necessitated by changes in environmental conditions need to be reflected in the systems. For example, facing changing demands of the market, the manufacturer may decide to outsource the production of some parts rather than making them in-house. The Acquisition WFMS needs to support such dynamically changing workflows to accommodate the changing of the business processes.

4 Traceability

Traceability refers to the ability to describe and follow the life of a conceptual or physical artifact, in both a forwards and backwards direction (i.e., from its origins, through its development, to its subsequent deployment and use, and through periods of on-going refinement

and iteration in any of these phases) (Ramesh and Jarke, 2000). The purpose is to help understand an artifact such as a workflow schema, from the beginning, through design, implementation, and maintenance. Göl and Finkelstein [11] highlight the differences between post and pre-specification traceability. Whereas the former is concerned about where a particular specification came from the latter is concerned about how it relates various artifacts are created to satisfy it. In the context of workflow systems, pre-specification traceability may correspond to the rationale behind the workflow schema specifications and post-specification traceability may correspond to workflow instantiations to satisfy a given specification.

Ramesh and Jarke [24] provide a detailed account of the various roles played by traceability in supporting complex organizational processes such as complex systems development.

Traceability can be seen as a measure of the quality of the systems development process and can facilitate communication among various stakeholders involved.

The need for traceability has been long recognized in complex design activities such as large scale software development. In fact, standards governing the development of complex, computer-based systems such as the MIL-STD-2167A (DoD, 1988) even mandate traceability. However, recent studies observe that there is wide variation in the quality and usefulness of traceability information among system development efforts. This is due in large part to the absence of clear guidelines, in either the standards or the current literature, on what traceability information must be captured and how it should be used. Recent research also suggests that the lack of pre-specification traceability is another important reason for problems associated with current traceability efforts [11]. Recent research fills this void and has created several reference models that can be used to create context specific traceability schemes to support varied stakeholder needs.

5 DWFM – Need for knowledge management

In this section, we discuss the need for augmenting current workflow management systems with the ability to represent process knowledge about their development and use. We present a snapshot of a case study in a manufacturing industry to highlight critical problems faced in developing and maintaining workflow systems. In the next section we outline our approach to addressing these problems using a traceability based knowledge management solution.

Consider the workflow to automate the manufacturing control process shown in Figure 1. This includes a workflow component that shows the tasks in manufacturing parts. After obtaining orders, the manufacturing management (MM) enquires material controlling (MC) department (1). MC gets BOM (Bill of Material) from DCC (2,3). MC enquires inventory for raw materials (4,5), and reports to MM (6). MM notifies the process-planning task (7). Using the process-plan and raw materials products are produced. Qualified products are sent to the product inventory (11,12,13).

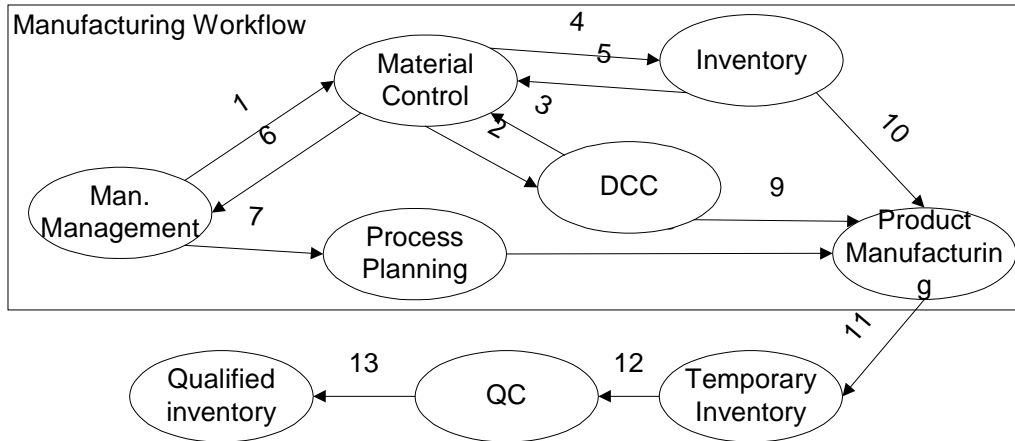


Figure 2: Manufacturing Control Process Workflow

To accommodate changes in business needs, the organization decides to outsource production of some parts. The manufacturing control process workflows need to be modified to accommodate this change. First, a new workflow to handle is created. The outsourcing workflow, shown in Figure 2 represents this revised schema. The manufacturing management (MM) sends orders to outsourcing companies. At the same time, DCC sends the parts requirements, and inventory provides raw materials to them. After the parts are produced, they are sent to product inventory. To accommodate such a change workflows that need to be redirected both in the model specification, and in workflow schema are identified. Finally, dependencies among workflows need to be maintained to avoid inconsistencies as well as to ensure correctness of the changed specifications.

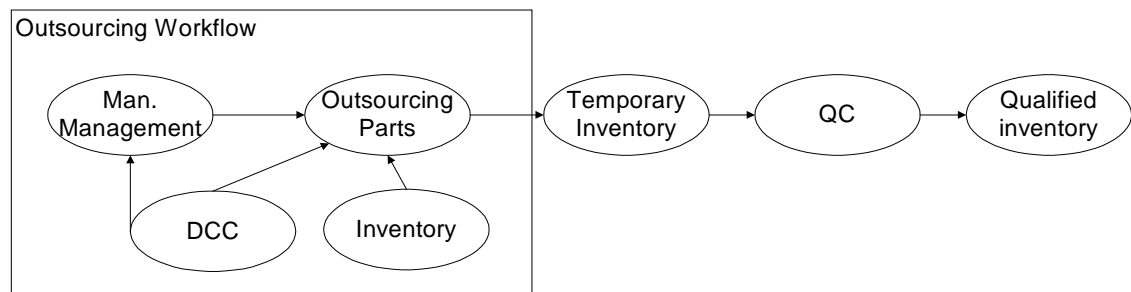


Figure 3: Workflow with Outsourcing

The above scenario highlights a variety of issues that need to be addressed in accommodating such changes. We discuss three critical issues here.

First, as organizations are facing continuous changes, workflows have to be modified correspondingly to reflect the changes in business processes they support. The need for

documentation of such changes in workflows have been emphasized in workflow literature [12]. It is argued that the main purpose of documentation in the context of workflow is human understanding. Users need to be able to understand what systems mean in business terms and what requirements does the systems satisfy. The documentation should also be able to suggest the effect of any proposed change. As business-process descriptions are typically mapped to workflow schemas in one big leap [13], interpreting and mapping is often time consuming and very difficult. All the design decision made during the process are typically not documented and the rich process knowledge behind these mappings these is often lost forever. We argue that this mapping must be explicit and fine-grained enough that it is not only possible to understand where a workflow segment came from.

Second, the lack of appropriate process knowledge may result in not only poor understanding of the workflow processes, but also hinder the ability of the organization to respond quickly to changing environmental conditions. For example, in the absence of detailed mapping between business needs, processes and workflows, it will be difficult or impossible to fully evaluate the repercussions of changes needed in the workflow systems to address changes in the environment. This problem is particularly acute dynamic business contexts where workflows are modified frequently. Often rationales for current processes and workflows have to be re-constructed before changes can be made. This process is prone to error and the potential for creating designs that may conflict with original design policies is very high.

Third, workflow re-configurations may not be reliable and consistent [8]. Re-configuration can happen to either the workflow specifications or executing workflow instance directly. While re-configuring workflows, it is important to identify all the dependencies among workflow fragments that are changed to identify and avoid inconsistencies. Since current workflow management systems have serious limitation in both the workflow model and the workflow execution environment with respect to failure detection [8], facilities to guarantee of workflow correctness can be very valuable. Sometimes, re-configurations are only partially done, leading to serious consequences For example, in our case study, the cost accounting workflows located in the financial module will be affected by changes made in the manufacturing process. After parts/products are manufactured, and before sent to the temporary inventory, the workflow system will trigger the task that calculates unit cost of production. However, when the manufacturing process control workflow is modified to support outsourcing, maintenance engineers may fail to detect changes necessary in other workflows such as the cost accounting workflow. Then, the unit cost of the product computed by this system will be erroneous.

6 Our Approach

In this section, we propose an approach to managing process knowledge in the development and maintenance of workflow systems and their instantiations using traceability. Traceability provides the ability to cross-reference various physical and conceptual artifacts produced in an organizational process such as workflow specification and execution. Our approach is based on the premise that a variety of tools can be developed to support the specification, maintenance, and instantiation of workflows if a rich history of their development can be captured using a

traceability scheme. This raises the question: what are the components that need to be traced in such a history. We have developed a conceptual model

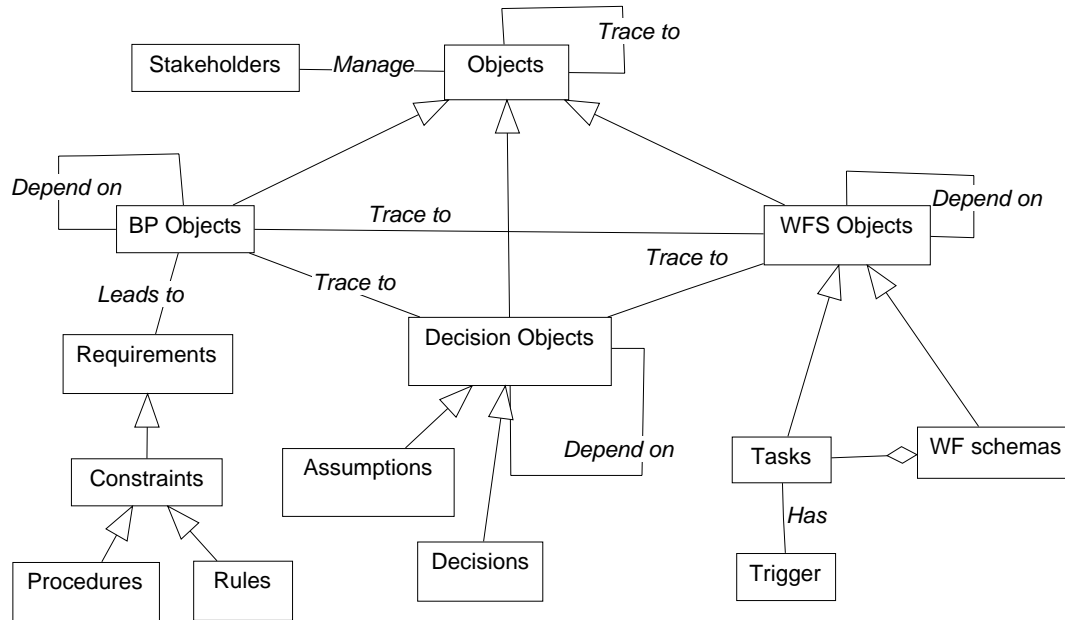


Figure 3. Conceptual Model

to address this question. Figure 3 shows the conceptual model that identifies important elements in a traceability scheme to represent the process knowledge in workflow system design, maintenance and implementation. The model is a specialization of reference models for traceability developed in [24] to represent complex organizational processes. The model consists of three major types of objects among which traceability is maintained, viz, Business Process Objects (BP objects), Decision Objects and Workflow System objects (WFS objects). The model also explicitly identifies the stakeholders involved in management (creation, modification and maintenance) of these various kinds of objects.

To build a workflow management system, business process model must be derived from business requirements. The need for supporting business processes lead to a set of requirements that must be satisfied by a workflow system. These requirements may be in the form of constraints (say, organizational procedures and rules that must be followed) as well. Workflow schemas and tasks that they are composed are examples of WFS objects. The model explicitly represents the traceability links between BP objects and WFS objects. For example, the links between a workflow schema or a task to a constraint or a specific requirement that leads to its creation can be explicitly represented using this scheme.

Business process objects and workflow objects can be represented in more detail to support their complete specification. For example, various tasks, dependencies among them, and the order of tasks can be specified in such a scheme. For each task, the stakeholders that are involved in the definition and execution can be specified. Further, triggers for each task may be

specified using rules of the form $T = \langle E, C, A \rangle$ [19], where E represents a set of events that can cause the activation of the trigger, C represents the conditions evaluated when active the trigger, and A represents a set of action performed if the events satisfy the conditions defined in C.

During the process of definition of workflows, various components get specified, modified, and elaborated. This process may consider many design alternatives. After analyzing and evaluating, some alternatives are rejected. During this process, the rationale and assumptions behind critical decisions can be captured.

By linking BP objects, workflow objects and decision objects, the history of each can be easily traced. When change need to be made in any component, the repercussions of changes can be easily ascertained. Further, potential conflicts that may arise due to intended changes could be easily identified.

The framework described here consists of elements to represent the actors, inputs and outputs of the workflow system development process as well as the linkages among them. It can be used to represent the following dimensions of traceability information.

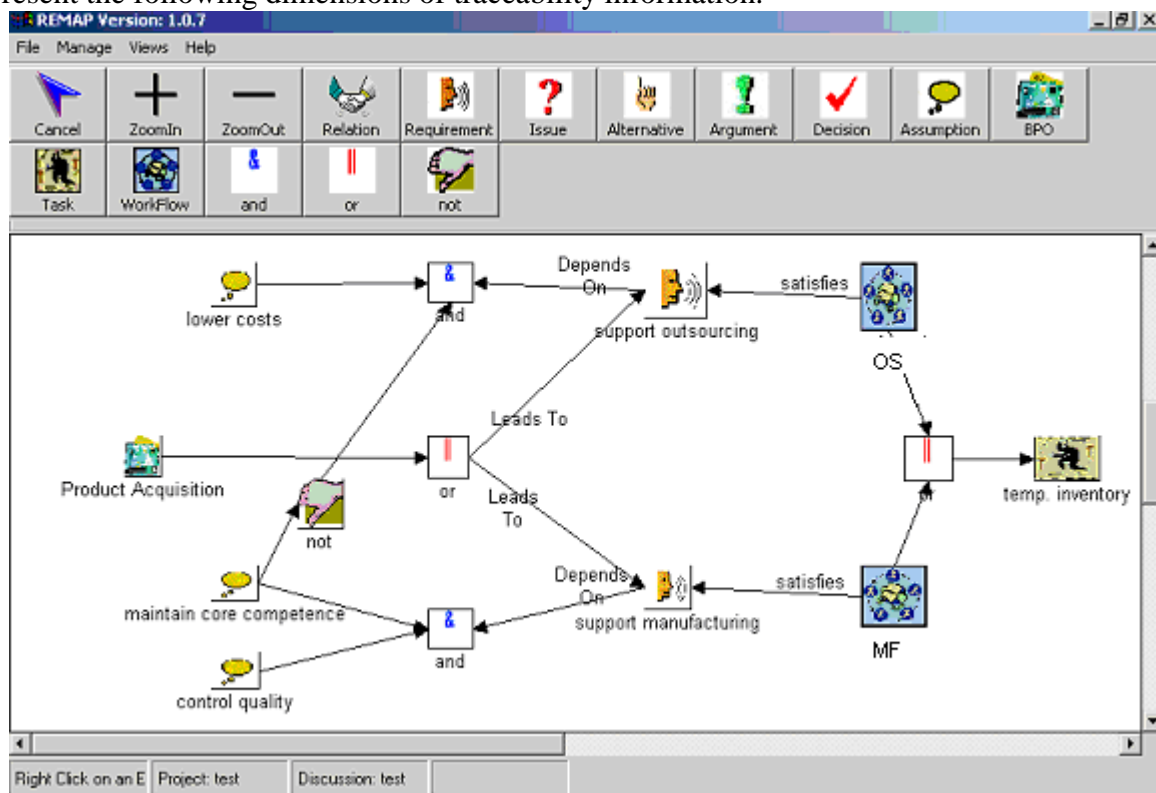


Figure 4. Tracing Acquisition workflows to context

- **What** information is represented - including salient attributes or characteristics of the information?
For example, links among the inputs and outputs of the customization process such as *Requirements*, *Assumptions*, *Designs*, *System components*, *Decisions*, *Rationale*, *Alternatives*, *Critical Success Factors*, etc. must be maintained. These represent the major conceptual elements among which traceability is maintained during the various stages of workflow systems development.
- **Who** are the stakeholders that play different roles in the creation, maintenance and use of various conceptual objects and traceability links across them?
- **Where** it is represented - in terms of sources that “document” traceability information.
- Examples of *sources* traceability information include business process objects such as requirement specification documents, meeting minutes, design documents, memoranda, telephone calls as well as references to various stakeholders.
- **How** this information is represented - both by formal and informal means and by how it relates to other components of traceability?
- **Why** a certain conceptual object was created, modified, or evolved?
The rationale behind the creation, modification, and evolution of various conceptual and physical objects need to be maintained. This information may include issues, alternatives, and arguments supporting and opposing various implementation decisions.
- **When** this process-related information was captured, modified, and evolved.

7 Supporting Acquisition WFKMS with Traceability

We now discuss the use of the above framework in a prototype traceability knowledge management systems created to support Acquisition WFMS. This system is capable of supporting the instantiation of the various elements in our conceptual model through a graphical interface. The system is intended to be used by various stakeholders involved in the creation and maintenance of workflow schemas and the business processes they support. We describe the functionalities of the system through three short scenarios corresponding to the ones described in section 6. The scenarios will address how the functionalities can alleviate the problems

earlier.

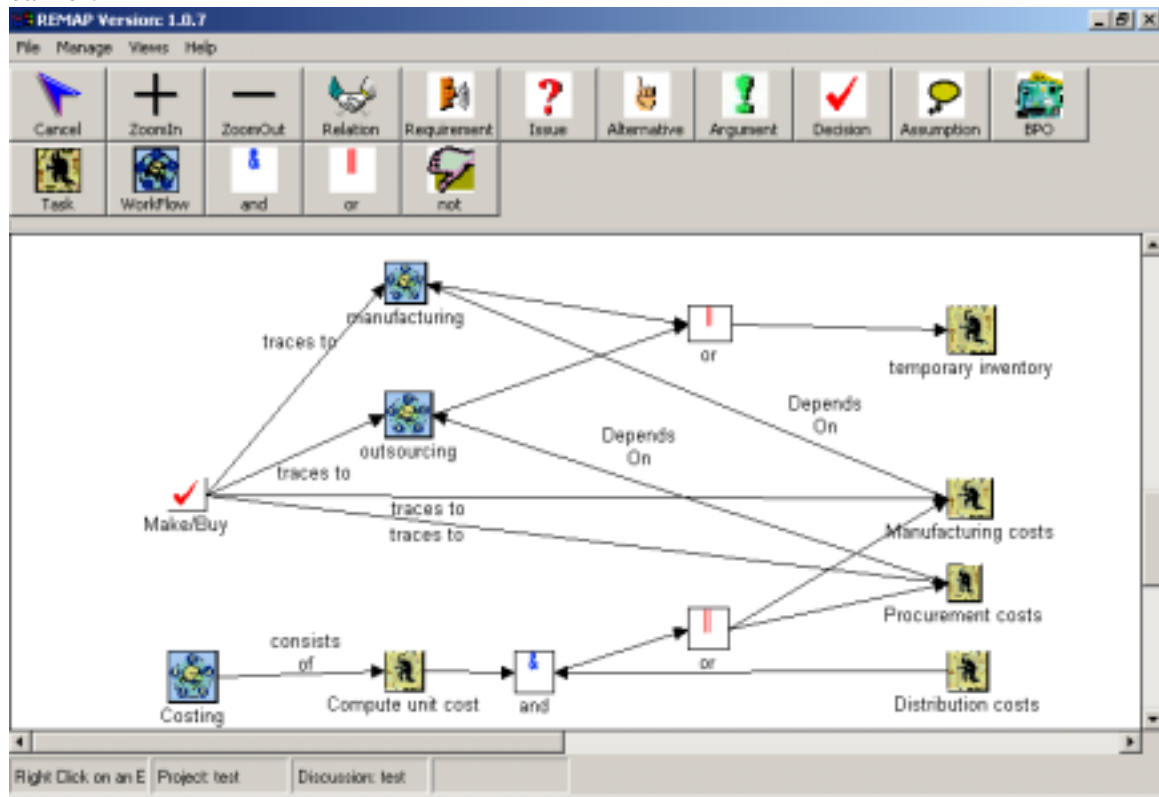


Figure 5.Managing dependencies

7.1 Traceability to context

First, we describe how the context in which a workflow schema is specified can be linked to it. Figure 4 shows a fragment of traceability knowledge that can be established between requirements and workflow objects. Returning to our example on the definition of workflows for manufacturing process control, observe that this workflow was guided by a requirement to support manufacturing. The requirement to either support manufacturing or support outsourcing are necessary to support the business process related to product acquisition. The option to support manufacturing is based on the assumptions that it helps control quality and maintain the core competencies of the organization. The later also objects to supporting outsourcing. However, outsourcing is supported by the assumption that it may lead to lower costs.

The scenario illustrates that a rich context in which a workflow is defined can be captured at varying levels of detail using our approach. This helps in improving the understanding of the workflow as well as its maintenance.

7.2 Dynamic Re-configuration

An important use of traceability knowledge is in the reconfiguration of workflows with changed business environment and processes. Recall from our case study that the organization may be forced to move from manufacturing to outsourcing due to changes in

the environment. In our example, the change in the validity of the assumption that outsourcing the product may result in losing core competencies and the validity of the assumption that it is cheaper to do so may suggest moving to outsourcing rather than manufacturing. The dynamic reconfiguration of the workflows from that shown in Figure 1 to that shown in Figure 2, in this instance, can be supported by our system with just changes in the validity of these assumptions. Our system uses a reason maintenance system to propagate the effects of changes in one component of process knowledge onto other. In this specific example, the system can suggest that the changed context should involve executing workflows specified in Figure 2.

7.3 Maintaining Integrity

Another important problem identified in our analysis is the difficulty in maintaining the integrity of workflows across organizational and task boundaries. For example, in Figure 5, a change in the manufacturing control process to include outsourcing has implications for workflows in other subsystems such as the cost accounting system in the financial module. Recall that the unit cost computation in this module assumes that the product is produced in-house. However, with the changes in the business process to move to outsourcing this assumption gets invalidated. The make/buy decision made in the manufacturing context, in effects, affects the workflows for computing unit costs. Our system provides the ability to manage such dependencies as well. When the workflows in the manufacturing control process change the system will prompt the user about potential conflicts that could arise. In the event that the repercussions can be formally modeled, the system can also suggest a reconfigured workflow for the cost accounting process.

In the brief scenarios described here, the capabilities of the system to address three key issues associated with workflow specification and maintenance are illustrated.

8 Related Work

Extensive research has been done on dynamic change within workflow systems. Maurer et al. [14] had designed MILOS system to support dynamic coordination of distributed software development teams by integrating project planning and workflow technologies over the Internet. It uses a workflow engine to monitor and implement all changing events. Shrivastava et al [15] proposed the Reliable Workflow Systems (RWS) based on CORBA, which uses task controllers to manage dynamic reconfigurations. Criticizing the RWS's tightly coupled system structure, Tari et al. [8] proposed a framework based on CORBA using even-channels. Weske [16] studied the ability of WFMS to adapt the structure of running workflow instances to modified workflow schemas.

New research issues come up with the proliferation of studies done on dynamic workflow management systems. Dynamic configuration involves either the work specification or the executing workflow instance. Changes to procedures, performed in an ad-hoc manner, can cause inefficiencies, inconsistencies, and catastrophic breakdowns [7], [8]. Ellis et al. [7] attempted to solve part of this problem by introducing a mathematical formalism to model and analyze dynamic structural changes within workflow systems. Reichert et al. (1997) presented a framework for the support of ad hoc structural changes of WFMSs. Those approaches are based on the knowledge of workflow schemas, rules, and

constraints. With the traditional design document, identifying relevant workflow schema and constraints on the flow is a time-consuming job.

Apart from problems posed by structural changes, there are a variety of issues raised by changes that might conflict with the underlying business goals, rules, and regulations implicitly stored in workflow systems. Zhao [17] indicates that different types of knowledge are contained in organizational Acquisition WFMS. This includes process knowledge that describes tasks, roles, rules, and routes, institutional knowledge that describes business procedures and regulations, and environment knowledge that describes environmental factors such as government. That knowledge is built into Acquisition WFMSs when designing workflow schema, discussing alternatives, and solving arguments among different end-users. However, some of them are not clearly represented in current workflow system models. Without realizing the constraints, dynamic changes may conflicts with business processes.

The importance of process knowledge and the context in which the process knowledge is created has been well recognized in recent research. Kwan [18] examined several workflow modeling methods and concluded that the current process models aimed at supporting process enactment only support partial views of a process. Based on Curtis's [19] four perspectives in process models, Kwan [18] defines a process meta-model to represent process knowledge components such as goals and sub goals, and the links among them. Our work is similar in spirit that we provide mechanisms to capture the rich contextual knowledge in which workflow systems are built, maintained and enacted.. Several frameworks are proposed to support the usage of Acquisition WFMS as the basic infrastructure in the development of organizational knowledge management systems [20], [21]. However no work has been done to study how to manage the processes and institutions' knowledge implicitly stored in workflow systems, especially in a dynamic environment.

Use of traceability to manage complex organizational processes has been explored in a variety of domains. For example, the importance of requirement traceability has been well documented in the software engineering literature. For example, REMAP system provides facilities to capture and reason with design rationale in domains ranging from concurrent engineering to new product development [22]. Pinheiro et al. describe TOOR (Traceability of Object-Oriented Requirement) system that uses multiple ways to trace requirements to support maintenance activities. PRIME [23] (Process Integrated Modeling Environments) framework empowers method guidance through process-integrated tools. The framework defines tool models, integrates tool models with method definition, interpret the environment and synchronize the tool and enactment mechanisms thereby supporting comprehensive traceability capture and use.

Ramesh and Jarke [24] present reference models for requirement traceability based on an extensive empirical study. These reference models not only provides a link from initial requirements to the actual system components, but also cover detailed traceability in various systems engineering activities such as requirement management, design allocation model, and compliance verification model.

We are currently developing a traceability knowledge management system to support Acquisition WFMS. XML will be used to represent ‘documents’ containing traceability knowledge. For example, XLink [25] will be used to represent links between the elements in the conceptual model. XLink provides a framework for creating both basic unidirectional links and more complex linking structures. It allows XML documents to assert linking relationships among more than two resources, associate metadata with a link, and express links that reside in a location separate from the linked resources. Xlink can not only represent association between two components (XML files), but also can capture knowledge of the link. Our environment is intended to provide a variety of mechanisms to support the various tasks involved in workflow management and maintenance. Further, we are investigating mechanisms for non-intrusive capture of traceability information to reduce the overhead involved. Finally, the empirical evaluation of the effectiveness of the proposed approach is also a subject of current work.

9 Discussion

The proposed approach to maintaining traceability requires careful consideration of several issues. First, the overhead involved in capturing detailed history including design rationale, assumptions, decisions, and constraints, can be very high. Second, though traceability can help improve correctness of dynamic re-configurations, a variety of factors such as the formalisms used in defining workflows, the architecture of the Acquisition WFMS components and the formality with which traceability knowledge is represented will determine the degree to which this process can be automated. Finally, the integration of traceability tools with workflow management systems may also pose interesting challenges.

Often organizations need to implement large and heterogeneous distributed systems where a set of interrelated tasks can be executed in efficient way. Workflow automation to coordinate activities throughout the enterprise is increasingly recognized as an important approach to support these requirements [1]. The WFMS can also play a vital role in managing knowledge of the enterprise, such as managing business rules [2]. As summarized in [3], the key benefits of WFMS are:

- Improved efficiency - automation of many business processes results in the elimination of many unnecessary steps
- Better process control - improved management of business processes achieved through standardizing working methods and the availability of audit trails
- Improved customer service – consistency in the processes leads to greater predictability in levels of response to customers
- Flexibility – software control over processes enables their re-design in line with changing business needs
- Business process improvement - focus on business processes leads to their streamlining and simplification

Proof of the ever-increasing interest in workflow management is the large number of commercial products that have appeared in the last few years including: Action Workflow System, of Action Technologies; IBM’s Flow Mark; Visual Workflow from

FileNet; OPEN/workflow, a WANG's product [4]. Workflow technology has become mainstream application-development tools and application-integration middleware that support the management of business process realization in a variety of application areas. Enterprise Resource Planning Systems (ERP) may be thought of as focusing on business applications of workflow management system [5]. ERP systems may be seen as "a set of more or less fine-grained business functions and a set of predefined workflows that realize complex business operations". When the system is customized for a specific setting, workflows can be modified and new ones can be constructed [6].

In the context of BPR (business process re-engineering), workflow management provides a means for enacting reengineered processes and for gathering information about the actual performance of these processes. As the consequence of rapidly changing environment, businesses need to reengineer business models and processes frequently. Those organizations in the modern business world, which refuse to change, are headed toward rapid obsolescence because they cannot compete. These business realities need to be supported by a WFMS by providing the ability to rapidly and dynamically change and enact workflows.

10 References

- [1] A. Sheth, "Workflow automation: applications, technology and research," presented at the 1995 ACM SIGMOD international conference on Management of data, 1995.
- [2] C. Moore, "Process knowledge," : www.e-workflow.org, 2000.
- [3] www.e-workflow.org, "The Key Benefits of Workflow," , vol. 2001: www.e-workflow.org, 2001.
- [4] G. Alonso and H. J. Schek , "Research issues in large workflow management systems," presented at the NSF Workshop on Workflow and Process Automation in Information Systems, Athens, Georgia, USA, 1996.
- [5] P. Hawking and B. McCarthy, "Industry Collaboration: A Practical Approach for ERP Education," presented at ACE 2000, Australia, 2000.
- [6] M.-T. Schmidt, "The Evolution of workflow standards," *IEEE Concurrency*, pp. 44-52, 1999.
- [7] C. Ellis and G. Rozenberg, "Dynamic change within workflow systems," presented at COOCS, CA, 1995.
- [8] Z. Tari and V. Pande, "Dynamic workflow management in CORBA distributed object systems," presented at International Symposium on Software Engineering for Parallel and Distributed Systems, 2000.
- [9] A. Sheth, "From contemporary workflow process automation to adaptive and dynamic workflow activity coordination and collaboration," presented at the 8th International Workshop on Database and Expert Systems Application, 1997.
- [10] S. Jajodia, "Interview: Amit Sheth on workflow technology," *IEEE Concurrency*, pp. 21-23, 1998.

- [11] O. C. Z. Gotel and A. C. W. Finkelstein, "An analysis of the requirements traceability problem," presented at the First International Conference on Requirements Engineering, 1994.
- [12] M. Jackson and G. Twaddle, *Business process implementation: Building workflow systems*: Addison-Wesley Press, 1997.
- [13] K. Meyer - Wegener and M. Bohm, "Conceptual workflow schemas," presented at International Conference on Cooperative Information Systems, 1999.
- [14] F. Maurer, B. Dellen, F. Bendeck, S. Glodmann, H. Jp;z, B. Kotting, and M. Schaaf, "Merging project planning and web-enabled dynamic workflow technologies," *IEEE Internet Computing*, vol. June, pp. 65-74, 2000.
- [15] S. K. Shrivastava and S. M. Wheeler, "Architectural support for dynamic reconfiguration of large scale distributed application," presented at the 4th International Conference on Configurabel Distributed Systems (CDS), Annapolis, 1998.
- [16] M. Weske, "Formal foundation and conceptual design of dynamic adaptions in a workflow management systems," presented at the 34th Hawaii International Conference on System Sciences, Hawaii, 2001.
- [17] J. L. Zhao, "Knowledge management and organizational learning in workflow systems," presented at AIS America Conference on Information Systems, Baltimore, Maryland, 1998.
- [18] M.-I. M. Kwan, "Process-oriented knowledge management," in *School of Management*. Boston: Boston University, 1999.
- [19] B. Curtis, M. I. Kellner, :, and J. Over, "Process Modeling," *CACM*, vol. 35, pp. 75-90, 1992.
- [20] M. Berger, E. Ellmer, and D. Merkl, "A learning component for workflow management systems," presented at 32th Hawaii International Conference on System Science-1998, Hawaii, USA, 1998.
- [21] C. Wargitsch and T. W. Felix theisinger, "An organizational-memory-based approach for na evolutionary workflow management -- concepts and implementation," presented at 32nd Hawaii International Conference on System Sciences, 1998.
- [22] B. Ramesh and V. Dhar, "Supporting Systems Development by Capturing Deliberations During Requirements Engineer," *IEEE Transactions on Software Engineering*, vol. 18, 1992.
- [23] K. Pohl and e. al., "PRIME - Toward process-integrated modeling environmnets," *ACM Transaction on Software Engineering and Methodology*, vol. 8, 1999.
- [24] B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *IEEE Transactions on Software Engineering*, vol. 27, 2001.
- [25] W3C, "XML Linking Language (XLink) Version 1.0," <http://www.w3.org/TR/xlink/>, 2000.

Managing process knowledge in acquisition of software systems

11 Background

A software development process consists of a set of activities needed to transform users' requirements into a software system [1]. It is a roadmap that helps create a timely and highly quality software product [2]. A process is defined as a set of high-level and/or low level activities, the order in which they are performed, roles of those involved in them, artifacts produced or processed by them, tools that support them, schedules, project plans and milestones [2, 3].

Software processes are the foundation for software engineering [2]. As important assets of organizations [4, 5], well-defined software processes provide the following benefits:

- Improve the performance, predictability and reliability of work processes;
- Increase productivity, and ensure the quality of final products.
- Make software development approach scalable, transferable, measurable and independent on particular people.
- Provide control of software development to ensure the outcomes of the development.
- Enable the development team consistently apply any development approach, which is especially important when implementing new software techniques.
- Enhance the communications between team members.
- Oriented new personnel.

Building processes from scratch each time would create high risks. The best way is to use tried and tested operations [13]. Hitchings identified two primary reasons for reuse of software development processes: increased effectiveness, and more time for enhancement [14]. Several standard software development processes reference models such as ISO/IEC 12207 [6, 7], IEEE/IEA 12207 [8-10], and RUP® [11, 12] provide the starting point in the definition software processes in organizations.. However, since the standards are developed to be broad in scope, they need to be tailored when applied to a specific project. The tailoring activities may include eliminating elements in the reference models, adding elements, and changing workflows.

In order to efficiently reuse process assets, practitioners need to understand the circumstances for which a process has been designed and the characteristics of the environment in which they are to be

used. Without this background knowledge, standard process models are too generic to readily applied in a software project. Recent literature suggests that standardized software processes are not followed rigourously, but elements of them are selected and tailored to suit project specific needs. Also, software processes are used differently in almost every project.. Therefore, in order for a software development team to reuse software process assets, it must understand the context in which they were applied so that their applicability to the current situation can be ascertained. Lack of contextual knowledge hinders practitioners' ability to understand processes and reuse them in other projects. Poor understanding and improper choice of software processes would result in inflated operation cycle time, and create conflicts in the development process [13].

This paper proposes a framework showing what contextual knowledge needs to be captured in software process tailoring. A research prototype that stores and retrieve software processes and related contextual knowledge is presented. In section 2, we discuss relevant research. The framework is presented in the section 3. The architecture of the prototype is discussed in section 4. Future work is discussed in section 5.

12 Related Research

Several standards and guidelines have been developed and published to help practitioners to construct or evaluate software processes. Standards such as ISO/IEC 12207 [6, 7], IEEE/IEA 12207 [8-10], and RUP® [11, 12] cover activities and tasks in various phase of software development. Each of these provides guidelines on how to tailor the standard for a project or an organization. However, the effectiveness of these guidelines has been mixed in efforts to tailor standards for a specific project or a domain. Demirors et al. [15] tailored ISO/IEC 12207 for an instructional software development. Polo et al. [16, 17] tailored the maintenance process in ISO/IEC 12207. Machado et al. [18] applied ISO/IEC 12207 and CMM model to improve a process for service development. These efforts highlight the challenges involved in process tailoring. An important observation from this research is that process definition is a knowledge intensive activities and capturing the knowledge about the definition of a process will help in tailoring and reusing a process in other projects.

Arguing that software processes are organizational valuable assets, Holdsworth proposed the reuse of proven processes instead of constructing them from scratch [13]. Avrilioni and Cunin [19] proposed

OPSIS approach to reuse process assets. OPSIS represents process components as a Petri-net. The selection of reusable process components mainly relies on the matching of component interfaces with the parameters. This approach can check the consistency of the process model structure, but ignores a variety of contextual knowledge used in process tailoring.

Henninger et al. [20, 21] presents an approach and a tool that supports the reuse of process fragments in constructing software development processes. Their system, called BORE, (Building an Organizational Repository of Experiences), abstracts contextual knowledge of a process fragments as rules. A decision support system guides users through a set of questions. Answers to the questions will help select process fragments. The resulting process is reviewed and necessary changes are made. Though this approach recognizes the importance of contextual knowledge, it fails to give clear guidelines on what knowledge needs to be captured in order to support reuse of process fragments. In summary, software development processes are valuable assets of organizations. Reusing proven processes can reduce risk and increase efficiency. However, to achieve such reuse, it is important to understand not software process elements but also the context in which they are used. This raises an important question: what are the elements of this contextual knowledge that will help in understanding and reusing processes? Our research is geared towards addressing this question and developing a decision system for the acquisition and use of this knowledge to support process reuse.

13 Framework

In this section, we propose a framework that defines the elements needed to be captured to support reuse process. The framework is shown in Figure 1. The elements include process chunks, products processed by process chunks, and different types of contextual knowledge that will help understand and reuse processes.

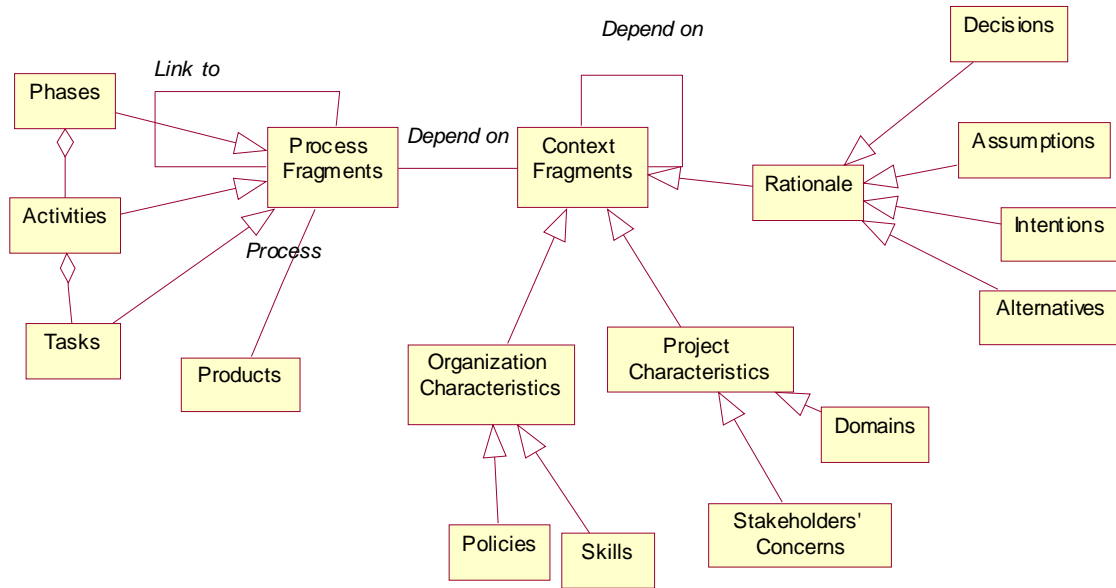


Figure 1. The framework

Three types of process fragments at different levels of abstraction are defined. The process chunks and their hierarchy are adapted from the Rational Unified Process (RUP®), a popular process model. The process fragment called Phase, refers to a stage such as requirement analysis, system design, and system implementation in software engineering processes. Each phase contains a workflow consisting of several activities, which are similar to “workflow details” in the RUP® model. Each activity will be refined to a set of tasks.

Besides three levels of process fragments, context fragments are defined. Context fragments contains contextual knowledge under which a process is constructed. Organizational characteristics such as organizational policies and developer skills are factors that may affect process construction. For example, developers’ lack of one skill may lead to elimination of the tasks that need such skill, or lead to adding training activities. Organizational policies may impose constraints on development efforts. The other factors that influence process constructions are project characteristics such as application domains and stakeholders’ concerns. Applications in one domain share common concerns, thus leading to some patterns in process configuration. For example, applications in the web development domain share commonalities such as short duration and emphasis on interface design. Common characteristics across projects suggest commonalities in software engineering processes [22]. Stakeholders’ concerns may impose constraints on software processes. For example, if stakeholders’ main concern is time-to-market, the process may omit or compress several activities. Both organization characteristics and project characteristics form the environment under which

decisions about processes are made. An important component of contextual knowledge are the rationale behind choice of process elements in a specific project. This includes decisions, assumptions, intentions and alternatives considered in the choice of a specific process fragment in a given project. These factors represent the reasoning processes of process engineers.. The organizational factors, project factors, and rationale can facilitate locating the right process for the right situation.

14 Knowledge Management System

In this section, we discuss a research prototype that can capture and use contextual knowledge defined in terms of the contextual elements described in Figure 1 to support process reuse. The research prototype is integrated with Rational Process Workbench®, an environment where process engineers can tailor the RUP® reference model for specific projects. The architecture of the prototype is shown in figure 2.

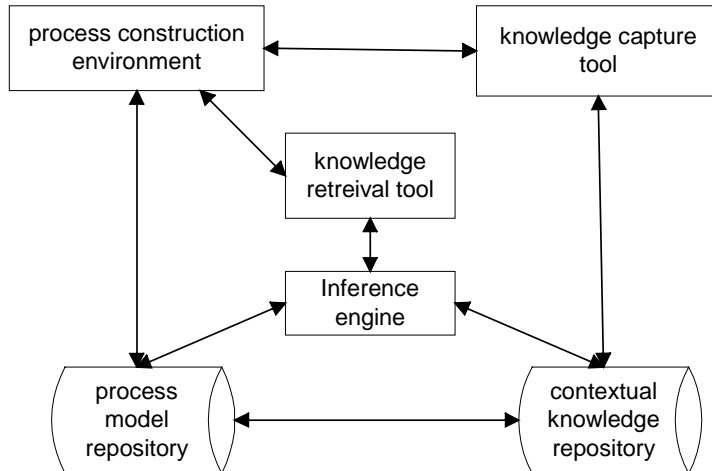


Figure 2. The architecture of the prototype

The prototype provide a tool for knowledge capture with which contextual knowledge can be defined and stored. The captured contextual knowledge is stored in a separate repository from process model. A knowledge retrieval tool guides users to locate proper process model or fragments by matching projects' characteristics with the history cases. An inference engine maintains the consistency of the knowledgebase of contextual knowledge and process models as they are incrementally defined and modified.

4.1 Acquiring Contextual knowledge

Users, working in their process construction environment such as Rational Process Workbench®, can invoke the knowledge capture tool. This tool provides facilities to define contextual knowledge fragments, relationships between different knowledge fragments, and relationships between knowledge fragments and process chunks. A process chunk can be a phase, a workflow diagram, an activity, a task, or a role.. The tool is implemented as a multi-user web enabled tool for different members of a project team to conduct conversations about the definition and use of process elements. As process engineers discuss and define process elements for a specific project, knowledge about the organizational and project characteristics as well as the rationale behind the choice of specific process fragments is captured and linked to relevant process fragments. Our environment provides an inference engine that maintains the consistency of the knowledge base that is evolving as these conversations incrementally define the knowledgebase.

In “contextual knowledge repository”, knowledge is stored at three levels—phase level, activity level, and task level. At the phase level, the knowledge about what phases are included, iterations of the phases, goals of each phase in each iteration, and reasons behind the decisions is stored. At activity level, the knowledge of implementation of each phase is stored. At the task level, the knowledge about implementation of each activity is stored.

4.2 Knowledge retrieval

The knowledge retrieval tool provides facilities to locate contextual knowledge and associated process fragments. There are three main steps in the process. At first, users provide general project related information such as details on the application domain and project characteristics. The inference engine matches this information with the knowledge stored at the phase level. In the first step, users can decide on what phases should be included, how many iterations are needed, and the goals for each iterations. After deciding on the high-level process model, users can refine their information so that inference engine can match activity level knowledge at the second step, and task level knowledge at the third step. The reusable process fragments at the activity level and task level can be located.

At each step, the inference engine could identify more than one matched fragment. Users can, then, review the fragments and related contextual knowledge to decide on their process design.

Conclusion

Our research proposes a framework that provides guidelines on the elements of contextual knowledge that need to be captured when tailoring software processes. Further, a prototype system to facilitate the acquisition and use of this knowledge in process tailoring has also been developed.

By integrating this tool with RUP Workbench®, a leading process construction tool, , we are currently evaluating the effectiveness of our approach using empirical studies of software developers engaged in process tailoring.

15 Related Publications

1. B. Ramesh and M. Jarke, "Towards Reference Models for Requirements Traceability", *IEEE Transactions on Software Engineering*, Vol 37, No 1, 2001.
2. A. Tiwana and B. Ramesh, "Supporting Information Product Development", *Decision Support Systems* , Vol. 31, 2001. pp. 241-262
3. A. Tiwana and B. Ramesh, "Integrating Knowledge on the Internet ", *IEEE Internet Computing* May/June 2001, pp. 32-39.
4. R. Baskerville, L. Levine, B. Ramesh, J. Pries-Heje and S. Slaughter, How Internet Companies Negotiate Quality", *IEEE Computer*, May 2001.pp. 51-57.
5. B. Ramesh, Process knowledge management with Traceability, *IEEE Software*, May/June 2002.
6. B. Ramesh, J. Pries-Heje, R. Baskerville, Internet Software Engineering: A different class of processes, *Annals of Software Engineering* (accepted for publication), 2002.
7. B. Ramesh and M. Kannan, Integrating group decision and negotiation support systems with work processes, In Proceedings of the Hawaii Intl. Conference on System Sciences, January 2001.
8. K. Surysekar and B. Ramesh, On managerial Incentives for process knowledge capture and use, In Proceedings of the Hawaii Intl. Conference on System Sciences, January 2001.
9. A. Tiwana and B. Ramesh, e-services: problems, opportunities and digital platforms, In Proceedings of the Hawaii Intl. Conference on System Sciences, January 2001.
10. K. Mohan and B. Ramesh, Supporting Knowledge Integration in Group Decision and Negotiation with Knowledge Traceability, in Proceedings of the International Conference of the ISDSS, Brunell, UK, July 2001.
11. B. Ramesh, K. Mohan, Supporting Information Product and Service Families with Traceability, In Proceedings of the Intl. Conf on Product Family Engineering, Bilboa, Spain, Nov. 2001.
12. P. Xu and B. Ramesh, Supporting Workflow Management Systems with Traceability, In Proceedings of the Hawaii Intl. Conference on System Sciences, January 2002.
13. K. Mohan and B. Ramesh, Managing Variability with Traceability in Product and Service Families, In Proceedings of the Hawaii Intl. Conference on System Sciences, January 2002.

14. K. Mohan and B. Ramesh, An ontology for variability in product families, In proceedings of the International Conference on Software Engineering International workshop on software product lines, Orlando, FL, 2002..

16 References:

1. Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. 1999: Addison-Wesley Longman, Inc.
2. Pressman, R.S., *Software Engineering: A Practitioner's Approach*. 2001, New York: McGraw-Hill Companies, Inc.
3. Rational, *Rational Unified Process*. 1999.
4. Arbaoui, S. and F. Oquendo. *Reuse sensitive process models: Are process elements software assets too?* in the 10th International Software Process Workshop. 1996.
5. Greenwood, R.M., et al. *An Asset View on the Software Process*. in the 10th International Software Process Workshop. 1996.
6. Singh, R., *INTERNATIONAL STANDARD ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES*. 1998.
7. Singh, R., *An Introduction to International Standard ISO/IEC 12207*. 1999.
8. IEEE/EIA, *Industry implementation of International Standard ISO/IEC 12207.2: 1995. (ISO/IEC 12207 standard for information technology - software life cycle processes - implementation considerations*. IEEE/EIA, 1998.
9. IEEE/EIA, *Industry implementation of International Standard ISO/IEC 12207.0: 1995. (ISO/IEC 12207 standard for information technology - software life cycle processes - implementation considerations*. IEEE/EIA, 1998.
10. IEEE/EIA, *Industry implementation of International Standard ISO/IEC 12207.1: 1995. (ISO/IEC 12207 standard for information technology - software life cycle processes - implementation considerations*. IEEE/EIA, 1998.
11. Rational, ., *Rational Unified Process*. 2002.
12. Rational ., *Rational Process Workbench*. 2002.
13. Holdsworth, J., *Software Process Design: out of the tar pit*: McGraw-Hill International (UK) Limited.
14. Hitchings, R. and M. Martinez. *Reuse of Process Elements-One Company's Experience*. in the 10th International Software Process Workshop (ISPW '96). 1996: IEEE.
15. Demirors, O., et al. *Tailoring ISO/IEC 12207 for instructional software development*. in *The 26th Euromicro Conference*. 2000.
16. Polo, M., et al. *MANTEMA: a Complete Rigorous Methodology for Supporting Maintenance based on The ISO/IEC 12207 Standard*. in *The Third European Conference on Software Maintenance and Reengineering*. 1999.
17. Polo, M., et al. *MANTEMA: a Software Maintenance Methodology Based on the ISO/IEC 12207 Standard*. in *Fourth IEEE International Symposium and Forum on Software Engineering Standards*. 1999.
18. Machado, C.F., L.C.d. Oliveira, and R.A. Fernandes. *Experience Report - Restructure of Processes based on ISO/IEC 12207 and SW-CMM in CELEPAR*. in *Fourth IEEE International Symposium and Forum on Software Engineering Standards*. 1999.
19. Avriilioni, D. and P.-Y. Cunin. *Process Model Reuse Support - The OPSIS Approach*. in the 10th International Software Process Workshop (ISPW '96). 1996.
20. Henninger, S. *Tools Supporting the Creation and Evolution of Software Development Knowledge*. in the 1997 International Conference on Automated Software Engineering (ASE'97). 1997: IEEE.
21. Henninger, S., *Turning Development Standards Into Repositories of Experiences*. *Software Process: Improvement and Practice*, 2001. 6.

22. Ward, S. and P. Kroll, *Building web solutions with the Rational Unified Process: unifying the creative design process and the software engineering process*. Rational Software White Paper, 1999.